Learning Partial Ordering Constraints from a Single Demonstration

Anahita Mohseni-Kabir, Charles Rich and Sonia Chernova
Worcester Polytechnic Institute
Worcester, MA, USA
(amohsenikabir|rich|soniac)@wpi.edu

ABSTRACT

Current approaches to learning partial ordering constraints by demonstration require demonstrating all (or almost all) possible completion orders. We have developed an algorithm that, for plans involving relative placement of objects, learns the partial ordering constraints from a single demonstration by letting the user specify naturally conceived reference frame information. This work is an example of a broader research agenda that involves applying principles of human collaboration to robot learning from demonstration.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms

Keywords

Learning from demonstration, collaboration

1. INTRODUCTION

This work is being conducted within the context of a larger project, one of whose goals is to build a bridge between research on robot learning from demonstration (LfD) [1] and research on human collaboration [5]. Specifically, we view learning as a type of collaboration—anyone who has been a teacher knows that both the learner and the teacher must have the shared goal of learning for the process to succeed. Furthermore, in a typical collaboration, both participants not only perform coordinated actions, they also *communicate* about the goals, subgoals, progress, problems, etc., as they arise.

One of the benefits we hope to achieve from this approach is to make it possible for humans to teach complex tasks to robots using only a small number of demonstrations. To do so, the human is expected to provide helpful advice along with the demonstrated actions, i.e., advice that can be naturally given by an untrained user. We are also working on, but not reporting here, how the robot can ask good questions, i.e., questions that an untrained user can easily answer.

We report here on a first example of this approach, which is an algorithm that, for certain types of actions, learns ordering constraints from a single demonstration.

2. THE PROBLEM

Learning ordering constraints is a classic problem in LfD. Demonstrations are intrinsically totally ordered, i.e., any set of (discrete, non-overlapping) actions performed in the real world are in a specific sequence. However, in many cases, only some of the demonstrated ordering is required. For example, even though you must demonstrate checking the air in all four tires of your car in some order, the order doesn't really matter. Learning the minimum required ordering constraints is important, because it makes the learned plan more flexible and reusable.

More formally, suppose your goal is to teach the following partially ordered plan, where A, B, C and D are primitive actions, $>$ means precedes and (B | C) means that B and C can happen in either order:

$$A > (B \mid C) > D$$

The usual approach [2, 3] to teaching this plan is to use two demonstrations, i.e., ABCD and ACBD, and then to unify these two total orders into a partial order. In the table-setting domain often used for research on this problem, A in this example could be placing a plate on the table, B and C could be placing the knife and fork on either side of the plate, and D could be placing a dust cover on the entire setting.

3. OUR APPROACH

But can we do better than requiring two demonstrations? Two demonstrations doesn't seem like so many in this example, but this approach does not scale well, since the number of possible completions of a partial order increases quickly as the number of steps increases. Would you need two demonstrations to teach this to a person? *Isn't it obvious to a person that the knife and fork can be placed in either order?*

The answer here is to think more deeply about the *reason* for the ordering constraints that are required. An ordering constraint is in general the symptom of an underlying *dependency* between actions. Let's see if we can make these dependencies explicit. To do so we need to digress briefly into the domain of numerical computations and data flow.

HRI'14, March 3–6, 2014, Bielefeld, Germany.
ACM 978-1-4503-2658-2/14/03.
http://dx.doi.org/10.1145/2559636.2559809.

Suppose that A, B, C and D are mathematical functions that are connected together with data flow as shown in Figure 1. Notice that A, B and C each have one input and one output; D has two inputs and one output. In this representation, it is obvious what the required ordering constraints are. B and C must happen after A because they each consume the output of A. D must happen after both B and C because it consumes each of their outputs. The order of B and C is unconstrained because there is no data flow between them.

So how can we represent a robotic demonstration, using actions such as PLACE, in this way? To start with, one input to PLACE should be the object to be moved. A second input should be a specification of the target location. How about outputs? In some programming languages, inputs to a subroutine that are *modified* (side effected) are also considered outputs. Let's view changing the location of an object as a side effect and therefore make the first input to PLACE also be an output.

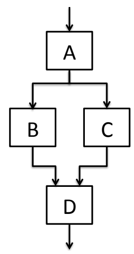

Figure 1:

Finally, and this is a key step, let's view every PLACE action as being performed in some *reference frame*, i.e., a coordinate system for the target location specification. In robot manipulation, there is always a coordinate system, either explicitly or implicitly. Furthermore, in many manipulation tasks, one object is often the reference frame for the placing of another object. For example, we place the fork "one inch to the right of the plate." Thus we add a third input to the PLACE action, which is the object that serves as the reference frame for this action.

Figure 2 shows the ABCD table setting demonstration represented using this formalization of PLACE. Notice that the plate output of the first action is the reference frame input for placing both the knife and the fork. The reference frame for placing the plate is the table, which does not move. Also, notice that we have added one additional artifact, which is a "virtual" action composing the plate, knife and fork into a composite object (the setting) that is the reference frame for placing the dust cover.

Once we have represented the demonstration this way, since Figure 2 is topologically identical to Figure 1, the required ordering constraints are directly learned from this *single* demonstration, namely:

Plate > (Knife | Fork) > Cover

4. CONCLUSION

Returning to our collaboration paradigm, what's going on here is that that the teacher is providing more information in the demonstration that just the final positions of the manipulated objects. Our formalization of the PLACE action requires that the reference frame be specified for every instance, e.g., that the plate is the reference frame for placing the knife.

Some researchers [2] have suggested trying to guess this reference frame information from heuristics about proximity of objects. However, from our point of view, this is information that a collaborating (communicating) user would very naturally provide as part of the demonstration.

For example, we imagine an LfD interface in which the user teleoperates a robot for demonstrations using an abstract GUI instead of a joystick or other direct control device. This GUI would have buttons for actions such as PLACE with associated menus to select target location options, such as "next to," followed by a choice of candidate reference frame objects. (There could also be a global default reference frame, such as the table.)

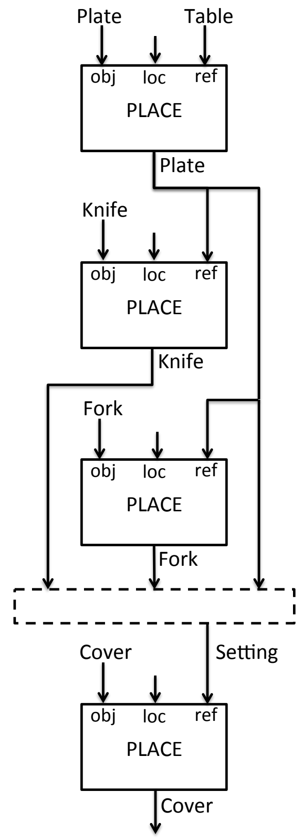

Figure 2: Demonstration

This is a very simple example of the style of collaborative interaction for LfD we are pursuing. We have implemented a system that performs the table-setting demonstration described above in simulation using the ANSI/CEA-2018 [4] standard for actions and plans.

Acknowledgement. This work is supported in part by the Office of Naval Research under Grant N00014-13-1-0735.

5. REFERENCES

[1] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57:469–483, 2009.

[2] S. Ekvall and D. Kragic. Robot learning from demonstration: A task-level planning approach. *Int. J. of Advanced Robotic Systems*, 5(3):223–234, 2008.

[3] M. Pardowitz et al. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(2):322–332, 2007.

[4] C. Rich. Building task-based user interfaces with ANSI/CEA-2018. *IEEE Computer*, 42(8):20–27, 2009.

[5] C. Rich, C. Sidner, and N. Lesh. Collagen: Applying collaborative discourse theory to human-computer interaction. *AI Magazine*, 22(4):15–25, 2001.